**Problem Statement**

**Product Dissection for Top Leading Platforms**

Welcome to this case study on dissecting and designing products for leading platforms. In this case study, you will delve into the intriguing world of schema design for a prominent platform of your choice. Your task is to choose a top-leading platform, research its features, and meticulously craft a schema design that encapsulates the essence of its functionality. Focusing on key entities, attributes, and relationships will give you invaluable insights into how data architecture drives the platform's effectiveness.

**Step 1: Choose a Leading Platform**

Select a leading platform that could span various domains such as social media, e-commerce, finance, or any other industry. This choice will form the foundation of your exploration into its schema design.

**Step 2: Research:**

Thoroughly research the platform you have selected. Investigate its core features, functionalities, and user interactions. Identify the top features that define its user experience and contribute significantly to its popularity.

**Step 3: Product Dissection and Real World Problems solved by the platform**

In this step, you will meticulously analyze the platform's standout features and how they provide innovative solutions to real-world challenges. By identifying key functionalities that resonate with users, you'll unravel how the platform effectively addresses problems and enhances user experiences. This dissection will serve as the foundation for understanding how the schema design aligns with the platform's core objectives.

**Step 4: Case Study on the real-world problems and approach to solving them**

In this pivotal step, you will expand on the real-world challenges uncovered in Step 3 through a comprehensive case study. Delve into specific instances where users encountered difficulties and showcase how the platform's unique features provided effective solutions. By dissecting the approach taken by the platform to overcome these challenges, you'll gain a deeper appreciation for the platform's user-centric design philosophy and how it shapes schema design.

**Step 5: Schema Design Based on Top Features**

Based on the features you have identified, craft a schema design that reflects the platform's data structure. Focus on the key entities, attributes, and relationships that underpin the chosen features. Your schema should capture the essence of how the platform organizes and utilizes its data.

**Step 6: Rationale Behind the Design**

While creating the schema design, consider the rationale behind the platform's choices. Reflect on why certain entities and relationships were chosen and how they align with the platform's goals. This will help you understand the strategic decisions driving the schema's architecture.

**Step 7: Create an ER Diagram**

Utilise tools like the Miro platform or similar applications to create an illustrative Entity-Relationship (ER) diagram. This diagram should vividly depict the entities, attributes, and relationships present within your schema design. The ER diagram will serve as a visual representation of your insights.

**Step 8: Presentation of Findings**

Present your findings in a clear and concise manner. Showcase your understanding of how the schema design impacts the platform's functionality and user experience. Explain how your chosen features are integrated into the schema and how the schema's structure supports the platform's objectives.

**Task Details:**

1. **Answer Submission:** Your submission should include well-structured solutions for all provided questions related to product schema designs.
2. **Video Creation:** Create an informative and engaging video where you thoroughly explain the Case Study.
3. **Depth and Clarity:** Ensure your solutions are detailed and showcase your understanding of product schema design principles. Similarly, the video provides clear explanations that are easy to understand for a wide audience.
4. **Creativity Encouraged:** You are welcome to utilize visuals, diagrams, or creative elements to enhance the clarity and impact of your explanations.

**Note:**

1. Duplicate this document and proceed to write your solutions and prepare your video.
2. Include the video link in this document before final submission.

Best of luck in completing this project and showcasing your prowess in dissecting and designing product schema for leading platforms!



### **Contribution - Individual**

GitHub Link -

### **Product Dissection for Ajio**

**Platform Chosen:** Ajio

**Domain:** E-commerce

AJIO is an Indian e-commerce company that operates as a subsidiary of Reliance Retail, which is part of Reliance Industries Limited. Here’s an overview of AJIO:

### **Overview:**

* **Parent Company**: Reliance Retail, part of Reliance Industries Limited.
* **Founded**: 2016.
* **Headquarters**: Mumbai, India.
* **Industry**: E-commerce, Fashion, Retail.

### **Key Features and Offerings:**

1. **Product Range**:
   * AJIO offers a wide range of products, including apparel, footwear, and accessories for men, women, and children.
   * It also provides a selection of curated and exclusive international brands, Indian ethnic wear, and indie (independent) labels.
2. **Fashion and Lifestyle**:
   * AJIO is known for its focus on fashion and lifestyle products, aiming to offer trendy and contemporary clothing.
   * It features an in-house label called AJIO Own, which focuses on stylish and affordable fashion.
3. **Exclusive Collaborations**:
   * AJIO collaborates with various national and international brands to offer exclusive collections.
   * It also partners with designers and influencers to bring unique styles to its customers.
4. **Technology and User Experience**:
   * AJIO employs a robust digital platform that provides a seamless shopping experience, including user-friendly navigation, easy returns, and multiple payment options.
   * The platform uses advanced technology to offer personalized recommendations and a smooth browsing experience.
5. **Customer Engagement**:
   * AJIO is active on social media platforms and engages with customers through various campaigns, promotions, and influencer collaborations.
   * It offers regular discounts, sales events, and loyalty programs to attract and retain customers.
6. **Logistics and Delivery**:
   * Leveraging the extensive network of Reliance Retail, AJIO provides efficient and reliable delivery services across India.
   * It offers various delivery options, including express delivery and standard shipping.

### **Market Position:**

* AJIO has established itself as a prominent player in the Indian e-commerce market, particularly in the fashion and lifestyle segment.
* It competes with other major e-commerce platforms such as Myntra, Flipkart, and Amazon in India.

### **Corporate Social Responsibility (CSR):**

* AJIO, through its parent company Reliance Retail, engages in various CSR activities focused on community development, education, and sustainability initiatives.

AJIO continues to grow by expanding its product range, enhancing its technological capabilities, and improving customer experience, solidifying its position as a key player in the Indian online retail market.

### **Step 1: Choose a Leading Platform**

**Platform:** Ajio  
**Industry:** E-commerce

### **Step 2: Research**

**Core Features of Ajio:**

1. **Product Catalog:** Wide range of products across different categories like clothing, accessories, and footwear.
2. **User Accounts:** Personalized user profiles for managing orders, wishlists, and preferences.
3. **Shopping Cart:** Functionality for users to add, remove, and manage items for purchase.
4. **Order Management:** Tracking and managing orders from placement to delivery.
5. **Recommendations:** Personalized product recommendations based on user behavior and preferences.
6. **Reviews and Ratings:** Users can rate and review products.
7. **Offers and Discounts:** Regular promotions, discounts, and coupon management.
8. **Payments:** Multiple payment options including credit/debit cards, net banking, and wallets.
9. **Customer Support:** Help and support for user queries and issues.

### **Step 3: Product Dissection and Real-World Problems Solved by Ajio**

**Key Functionalities and Solutions:**

1. **Product Catalog Management:** Allows users to browse through a vast selection of products, solving the problem of limited choices in physical stores.
2. **Personalized Experience:** User accounts and personalized recommendations enhance the shopping experience by catering to individual preferences.
3. **Efficient Order Management:** Simplifies the process of tracking and managing orders, providing transparency and convenience.
4. **Customer Feedback:** The reviews and ratings system helps other users make informed decisions and allows Ajio to maintain product quality.
5. **Cost Savings:** Offers and discounts provide users with cost-saving opportunities, making shopping more affordable.
6. **Seamless Payments:** Multiple payment options ensure a hassle-free checkout process, addressing different user preferences.
7. **Robust Customer Support:** An efficient customer support system addresses user queries and issues promptly, enhancing overall satisfaction.

### **Step 4: Case Study on Real-World Problems and Approach to Solving Them**

**Case Study Example:**

**Problem:** Limited product variety and difficulty in finding specific items in physical stores.

**Solution:** Ajio's extensive product catalog and advanced search/filter options.

**Approach:**

* Users can browse and filter products by category, brand, price range, and more.
* Personalized recommendations suggest products based on browsing and purchase history.
* User reviews and ratings provide insights into product quality and fit, aiding in decision-making.

**Case Study Example:**

**Problem:** Uncertainty about order status and delivery.

**Solution:** Comprehensive order management system.

**Approach:**

* Users receive real-time updates on order status, from placement to delivery.
* Tracking features allow users to monitor their package's journey.
* Automated notifications keep users informed about any changes or delays.

### **Step 5: Schema Design Based on Top Features**

**Entities and Attributes:**

1. **Users**
   * user\_id (PK)
   * username
   * email
   * password
   * date\_joined
2. **Products**
   * product\_id (PK)
   * name
   * description
   * price
   * category\_id (FK)
   * brand\_id (FK)
   * stock\_quantity
3. **Categories**
   * category\_id (PK)
   * name
4. **Brands**
   * brand\_id (PK)
   * name
5. **Orders**
   * order\_id (PK)
   * user\_id (FK)
   * order\_date
   * total\_amount
   * status
6. **Order\_Items**
   * order\_item\_id (PK)
   * order\_id (FK)
   * product\_id (FK)
   * quantity
   * price
7. **Reviews**
   * review\_id (PK)
   * product\_id (FK)
   * user\_id (FK)
   * rating
   * comment
   * review\_date
8. **Shopping\_Cart**
   * cart\_id (PK)
   * user\_id (FK)
9. **Cart\_Items**
   * cart\_item\_id (PK)
   * cart\_id (FK)
   * product\_id (FK)
   * quantity
10. **Payments**
    * payment\_id (PK)
    * order\_id (FK)
    * amount
    * payment\_date
    * payment\_method

**SQL Table Creation for the schema:**

-- Users table

CREATE TABLE Users (

user\_id INT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(100) NOT NULL,

date\_joined TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Products table

CREATE TABLE Products (

product\_id INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT,

price DECIMAL(10, 2) NOT NULL,

category\_id INT,

brand\_id INT,

stock\_quantity INT,

FOREIGN KEY (category\_id) REFERENCES Categories(category\_id),

FOREIGN KEY (brand\_id) REFERENCES Brands(brand\_id)

);

-- Categories table

CREATE TABLE Categories (

category\_id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

-- Brands table

CREATE TABLE Brands (

brand\_id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

-- Orders table

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

user\_id INT,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

total\_amount DECIMAL(10, 2) NOT NULL,

status VARCHAR(50) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

-- Order\_Items table

CREATE TABLE Order\_Items (

order\_item\_id INT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT NOT NULL,

price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

-- Reviews table

CREATE TABLE Reviews (

review\_id INT PRIMARY KEY,

product\_id INT,

user\_id INT,

rating INT CHECK (rating >= 1 AND rating <= 5),

comment TEXT,

review\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (product\_id) REFERENCES Products(product\_id),

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

-- Shopping\_Cart table

CREATE TABLE Shopping\_Cart (

cart\_id INT PRIMARY KEY,

user\_id INT,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id)

);

-- Cart\_Items table

CREATE TABLE Cart\_Items (

cart\_item\_id INT PRIMARY KEY,

cart\_id INT,

product\_id INT,

quantity INT NOT NULL,

FOREIGN KEY (cart\_id) REFERENCES Shopping\_Cart(cart\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

-- Payments table

CREATE TABLE Payments (

payment\_id INT PRIMARY KEY,

order\_id INT,

amount DECIMAL(10, 2) NOT NULL,

payment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

payment\_method VARCHAR(50) NOT NULL,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

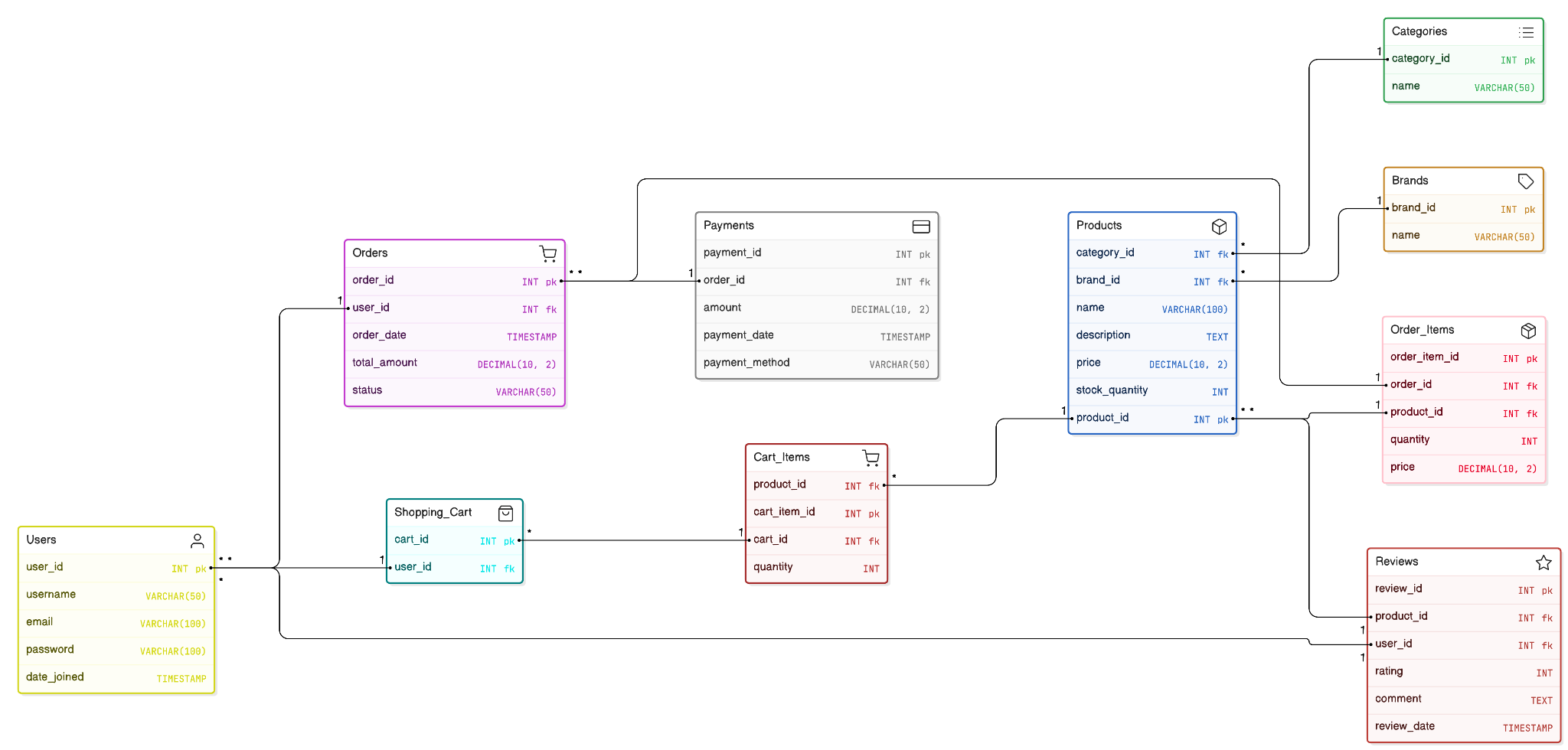
);

### **Step 6: Rationale Behind the Design**

* **Users:** Central entity for managing user-specific data including authentication and personalization.
* **Products, Categories, Brands:** Core entities to manage product information and organization.
* **Orders and Order\_Items:** Essential for tracking user purchases and order details.
* **Reviews:** Facilitates user feedback, enhancing the shopping experience for others.
* **Shopping\_Cart and Cart\_Items:** Manages user-selected items before purchase, improving the shopping process.
* **Payments:** Ensures secure and efficient transaction processing.

### **Step 7: Create an ER Diagram**

Let's construct an ER diagram that vividly portrays the relationships and attributes of the entities within the Ajio schema. This ER diagram will serve as a visual representation, shedding light on the pivotal components of Ajio’s data model.



### **Explanation of the ER Diagram**

This ER diagram represents the schema design for an e-commerce platform like Ajio, where each service handles specific functionalities related to the platform. Below is an explanation of the entities, their attributes, and the relationships between them:

#### **1. Users**

* **user\_id** (INT, PK): Unique identifier for each user.
* **Username** (VARCHAR(50)): The username chosen by the user.
* **Email** (VARCHAR(100)): The email address of the user.
* **Password** (VARCHAR(100)): The encrypted password of the user.
* **date\_joined** (TIMESTAMP): The date and time when the user created the account.

#### **2. Orders**

* **order\_id** (INT, PK): Unique identifier for each order.
* **user\_id** (INT, FK): References the user who placed the order.
* **order\_date** (TIMESTAMP): The date and time when the order was placed.
* **total\_amount** (DECIMAL(10, 2)): The total amount of the order.
* **status** (VARCHAR(50)): The current status of the order (e.g., pending, shipped, delivered).

#### **3. Payments**

* **payment\_id** (INT, PK): Unique identifier for each payment.
* **order\_id** (INT, FK): References the order associated with the payment.
* **amount** (DECIMAL(10, 2)): The amount paid.
* **payment\_date** (TIMESTAMP): The date and time when the payment was made.
* **payment\_method** (VARCHAR(50)): The method used for the payment (e.g., credit card, PayPal).

#### **4. Products**

* **product\_id** (INT, PK): Unique identifier for each product.
* **category\_id** (INT, FK): References the category to which the product belongs.
* **brand\_id** (INT, FK): References the brand of the product.
* **name** (VARCHAR(100)): The name of the product.
* **description** (TEXT): The description of the product.
* **price** (DECIMAL(10, 2)): The price of the product.
* **stock\_quantity** (INT): The available stock quantity of the product.

#### **5. Categories**

* **category\_id** (INT, PK): Unique identifier for each category.
* **name** (VARCHAR(50)): The name of the category.

#### **6. Brands**

* **brand\_id** (INT, PK): Unique identifier for each brand.
* **name** (VARCHAR(50)): The name of the brand.

#### **7. Order\_Items**

* **order\_item\_id** (INT, PK): Unique identifier for each order item.
* **order\_id** (INT, FK): References the order to which the item belongs.
* **product\_id** (INT, FK): References the product being ordered.
* **quantity** (INT): The quantity of the product ordered.
* **price** (DECIMAL(10, 2)): The price of the product at the time of the order.

#### **8. Shopping\_Cart**

* **cart\_id** (INT, PK): Unique identifier for each shopping cart.
* **user\_id** (INT, FK): References the user who owns the cart.

#### **9. Cart\_Items**

* **cart\_item\_id** (INT, PK): Unique identifier for each cart item.
* **cart\_id** (INT, FK): References the shopping cart to which the item belongs.
* **product\_id** (INT, FK): References the product added to the cart.
* **quantity** (INT): The quantity of the product added to the cart.

#### **10. Reviews**

* **review\_id** (INT, PK): Unique identifier for each review.
* **product\_id** (INT, FK): References the product being reviewed.
* **user\_id** (INT, FK): References the user who wrote the review.
* **rating** (INT): The rating given by the user (usually on a scale of 1 to 5).
* **comment** (TEXT): The review comment.
* **review\_date** (TIMESTAMP): The date and time when the review was posted.

### **Relationships**

* **Users ↔ Orders:** One-to-Many (A user can place many orders, but an order belongs to only one user).
* **Orders ↔ Payments:** One-to-One (Each order has one payment associated with it).
* **Orders ↔ Order\_Items:** One-to-Many (An order can contain many order items, but an order item belongs to only one order).
* **Users ↔ Shopping\_Cart:** One-to-One (Each user has one shopping cart).
* **Shopping\_Cart ↔ Cart\_Items:** One-to-Many (A shopping cart can have many cart items, but a cart item belongs to only one cart).
* **Products ↔ Order\_Items:** One-to-Many (A product can be part of many order items, but an order item refers to only one product).
* **Products ↔ Reviews:** One-to-Many (A product can have many reviews, but a review refers to only one product).
* **Users ↔ Reviews:** One-to-Many (A user can write many reviews, but a review is written by only one user).
* **Categories ↔ Products:** One-to-Many (A category can contain many products, but a product belongs to only one category).
* **Brands ↔ Products:** One-to-Many (A brand can have many products, but a product belongs to only one brand).

### **Explanation of Symbols**

* **PK:** Primary Key
* **FK:** Foreign Key
* \*\*1..\*: \*\* One-to-Many relationship
* **1..1:** One-to-One relationship

### **Step 8: Presentation of Findings**

#### **Schema Design Impact on Functionality and User Experience:**

* **User Personalization:** The Users entity, combined with Reviews and Shopping\_Cart, allows for a tailored shopping experience with personalized recommendations.
* **Efficient Product Management:** The Products, Categories, and Brands entities ensure organized and easy navigation through the product catalog.
* **Order Tracking:** The Orders and Order\_Items entities provide users with real-time order status updates, enhancing transparency and trust.
* **Feedback Mechanism:** The Reviews entity allows users to make informed purchase decisions based on the experiences of other shoppers.
* **Seamless Transactions:** The Payments entity supports various payment methods, ensuring a smooth checkout process.

### **Conclusion**

In this case study, we have explored the schema and ER diagram for Ajio, a leading Indian e-commerce company that operates as a subsidiary of Reliance Retail, which is part of Reliance Industries Limited. Ajio has transformed the online shopping experience, offering users a diverse range of fashion and lifestyle products while addressing real-world challenges in the E-commerce industry. This ER diagram provides a comprehensive view of the data architecture for an e-commerce platform like Ajio. By organizing the data into specific entities and defining their relationships, it supports the platform's functionality, including user management, product catalog, order processing, shopping cart management, payments, and customer reviews. This modular design enhances scalability, maintainability, and user experience.

**Video Link -** [**https://drive.google.com/file/d/1reA\_p5qSL5gPf7VRzAt00fT-TTX3-dDW/view?usp=sharing**](https://drive.google.com/file/d/1reA_p5qSL5gPf7VRzAt00fT-TTX3-dDW/view?usp=sharing)

**—-----------------------------------Thank You—---------------------------------------**